Erick Cabrera

ITM 311-02

Lab No. 3

September 11, 2016

The purpose of this lab was to create a program that calculated the maximum distance, height, and time of flight of a projectile when launched.

**Source Code:**

// the packages

import java.text.DecimalFormat;

import javax.swing.JOptionPane;

import java.lang.Math;

// the class definition

public class Projectile

{

// the global variables are declared

static double angle, distance, height, time, velocity;

final static double gravity = 32.2;

// the class constructor ( initialize values )

public Projectile()

{ angle = 0; distance = 0; height = 0; time = 0; velocity = 0; }

// the method to show object instantiation

public void ObjectInstantiated()

{

// display a message

JOptionPane.showMessageDialog(null, "object created!",

"Result", JOptionPane.PLAIN\_MESSAGE);

}

// the method to obtain data

public void getData()

{

// local variables declared and assigned initial values

String firstNum = "", secondNum = "";

// local variables updated via prompt boxes

firstNum = JOptionPane.showInputDialog("Enter initial velocity");

secondNum = JOptionPane.showInputDialog("Enter angle (degrees)");

// update the global variables

velocity = Double.parseDouble(firstNum);

// convert the angle from degrees to radians

angle = Double.parseDouble(secondNum) \* Math.PI / 180.0;

}

// method to compute maximum distance

public double computeMaxDistance(double v, double theta)

{

// declare some local variables

double maxDist = 0.0;

// assign values to the global variables

velocity = v;

angle = theta;

// perform the computations

maxDist = Math.pow(velocity, 2.0);

maxDist \*= Math.sin(2 \* angle) / gravity;

// return a value

return maxDist;

}

// the main() method

public static void main(String args[])

{

// introduce a DecimalFormat object

DecimalFormat twoPlace = new DecimalFormat("0.00");

// declare some variables that are local to main

double myVelocity = 0, myDistance = 0, myAngle = 0;

// introduce a class object

Projectile p = new Projectile();

// announce the object

p.ObjectInstantiated();

// object calls the getData() method

p.getData();

// assign globals to the locals

myVelocity = velocity;

myAngle = angle;

// object calls the computeMaxDistance() method

myDistance = p.computeMaxDistance(myVelocity, myAngle);

// display the result in a message box

JOptionPane.showMessageDialog(null, "maximum distance: " +

twoPlace.format(myDistance) + " feet ", "Result",

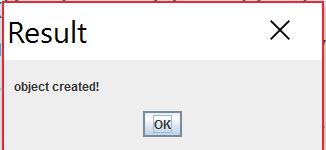
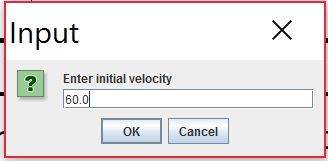
JOptionPane.PLAIN\_MESSAGE);

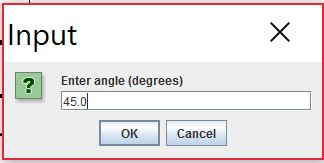
System.exit(0);

}// end the main() method

}// end the class definition

**Output:**





**Modified Source Code:**

/\* Program to calculate maximum distance, height,

\* and time of flight of projectile.

\* Programmer: Erick Cabrera, File Name: Projectile.java

\*/

// the packages

import java.text.DecimalFormat;

import javax.swing.JOptionPane;

import java.lang.Math;

// the class definition

public class Projectile

{

// the global variables are declared

static double angle, distance, height, time, velocity;

final static double gravity = 32.2;

// the class constructor ( initialize values )

public Projectile()

{ angle = 0; distance = 0; height = 0; time = 0; velocity = 0; }

// the method to show object instantiation

public void ObjectInstantiated()

{

// display a message

JOptionPane.showMessageDialog(null, "object created!",

"Result", JOptionPane.PLAIN\_MESSAGE);

}

// the method to obtain data

public void getData()

{

// local variables declared and assigned initial values

String firstNum = "", secondNum = "";

// local variables updated via prompt boxes

firstNum = JOptionPane.showInputDialog("Enter initial velocity");

secondNum = JOptionPane.showInputDialog("Enter angle (degrees)");

// update the global variables

velocity = Double.parseDouble(firstNum);

// convert the angle from degrees to radians

angle = Double.parseDouble(secondNum) \* Math.PI / 180.0;

}

//method to compute maximum distance

public double computeMaxDistance(double v, double theta)

{

// declare some local variables

double maxDist = 0.0;

// assign values to the global variables

velocity = v;

angle = theta;

// perform the computations

maxDist = Math.pow(velocity, 2.0);

maxDist \*= Math.sin(2 \* angle) / gravity;

// return a value

return maxDist;

}

//method to compute maximum height attained

public double computeMaxHeight(double v, double theta){

//declare local variables

double maxHeight = 0.0;

//assign values to global variables

velocity = v;

angle = theta;

//perform the computation

maxHeight = (Math.pow(velocity, 2) \* Math.pow(Math.sin(angle), 2))/(2 \* gravity);

//return a value

return maxHeight;

}

public double computeTimeOfFlight(double v, double theta){

//declare local variables

double time = 0.0;

//assign values to global variables

velocity = v;

angle = theta;

//perform the computation

time = (2 \* velocity \* Math.sin(angle)/gravity);

//return a value

return time;

}

// the main() method

public static void main(String args[])

{

// introduce a DecimalFormat object

DecimalFormat twoPlace = new DecimalFormat("0.00");

// declare some variables that are local to main

double myVelocity = 0, myDistance = 0, myAngle = 0, myHeight = 0, myTime = 0;

// introduce a class object

Projectile p = new Projectile();

// announce the object

p.ObjectInstantiated();

// object calls the getData() method

p.getData();

// assign globals to the locals

myVelocity = velocity;

myAngle = angle;

// object calls the computeMaxDistance() method

myDistance = p.computeMaxDistance(myVelocity, myAngle);

//object calls the computeMaxHeight() method

myHeight = p.computeMaxHeight(myVelocity, myAngle);

//object calls the computeTimeOfFlight() method

myTime = p.computeTimeOfFlight(myVelocity, myAngle);

// display the result in a message box

JOptionPane.showMessageDialog(null, "maximum distance: " +

twoPlace.format(myDistance) + " feet " + "\n"

+ "maximum height attained: " + twoPlace.format(myHeight)

+ " feet " + "\n" + "time of flight: " +

twoPlace.format(myTime) + " seconds ", "Result",

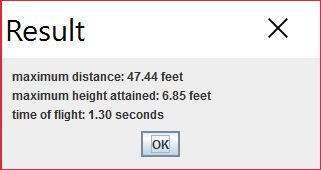
JOptionPane.PLAIN\_MESSAGE);

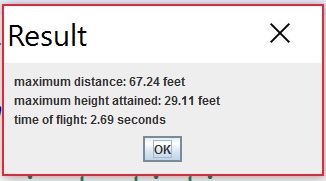
System.exit(0);

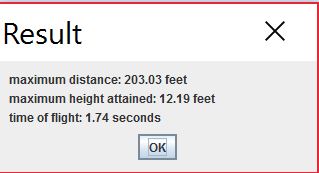
}// end the main() method

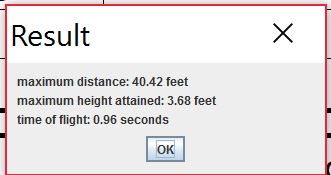
}// end the class definition

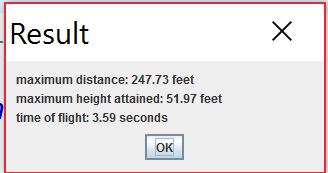
**Modified Output:**

A)

B)

C)

D)

E)